Architecture Support for Data Isolation & Memory Monitoring

Arrvindh Shriraman, Sandhya Dwarkadas, and Michael L. Scott

Department of Computer Science, University of Rochester

**Motivation**
- Multi-core processors based on shared memory programming will soon dominate the computing spectrum
- Coordinating and synchronizing data shared across multiple threads is hard!
- Tracking memory location accesses is difficult because of transparent coherence events
- Cannot issue speculative operations to memory because hardware protocol does not support undoing of writes

**Shared Memory ++**
- Memory Monitoring (MM)
  - provides read/write access summaries of cache blocks
  - event-notification of desired coherence events
  - Apps: Reliability, Security, Watchpoints, and Debugging
- Data Isolation (DI)
  - allows control over propagation of writes to remote threads
  - buffer written locations and commit or undo as an atomic unit
  - Apps: Sand-boxing, Transactional programming, Speculation

**DIMM Hardware Support**
- Decoupled hardware primitives for DIMM help
  - refine architecture incrementally
  - software evolve the API and use in various applications
  - decouple policy from mechanism
- Memory Monitoring primitives
  - Alert-On-Update: precise but bounded size
  - Signatures: imprecise but unbounded
  - CST: track inter-processer conflicts for all watched locations
- Data Isolation primitives
  - PDC: private cache
  - Redo-Log: holds cache overflows in virtual memory

**Data Isolation**
- Lazy Coherence
  - Caches detach lines selectively from coherence protocol
  - track coherence messages and choose time to enforce rules
  - TM buffers TStores; TI allows incoherence with remote TM
  - TM allows concurrent sharers & isolates data in cache
  - TMI & TI just require a flash-clear to convert lines to MESI
- Redo-Buffer
  - A per-thread hash-table in virtual memory
  - Hardware controller
    - fills table with "TM" write-back data blocks
    - performs look-aside transparently on L1 misses

**Conflict Summary Tables (CST)**
- Record inter-processor R-W, W-W & W-R conflicts
- Decouples access conflict tracking from access tracking

**Memory Monitoring**
- Alert-On-Update (AOU)
  - New instruction, Alouid, sends marks cache line
  - A-tagged line on invalidation jumps to handler
    - trigger event type can be capacity eviction or coherence
  - Extension to AOU: TMI, TI, and TMI/TI actions

**Access summary Signatures**
- Insert addresses accessed by thread in hardware bloom filters. (Reads update Rm & Writes update Wm)
  - unboundedness, decouples tracking from caches
  - Special instructions access cache blocks and insert physical address into bloom filter
  - Coherence requests snoop signatures, test for membership and piggy-back conflict type on response message

**Data Conflicts**
- Fastpath Transactions
- Overflow Transactions

**Hardware-acceleration of Software-controlled transactions**
- DIMM aids improve software-controlled TMs
  - FlexTM: Curry's Processor
  - Lazy encourages progress

**Conclusion**
- Data-Isolation and Memory-Monitoring primitives will help multi-core chips achieve widespread use across traditional and emerging application domains
- Decoupling the hardware components will help refine the architecture incrementally and help software evolve the API
- Use simple hardware to accelerate the common case, minimize hardware state and employ software for the uncommon case

**Other Uses**
- Synchronization: fast mutexes and asynchronous messages
- Debugging: watchpoints and race detectors
- Security: buffer overflow attacks, information-flow trackers & debugging
- Speculation: thread-level speculation and lock elision

Web: http://www.cs.rochester.edu/research/cosyn/

---

**FlexTM [ISCA'08]**
- FlexTM deploys
  - Signatures for detecting and notifying conflicts
  - CSTs for managing conflicts
- Lazy caches for in-cache data isolation and Redo-Buffer for handling cache overflows
- AOU for propagating abort events to remote transactions
- FlexTM software
  - checkpoints register at Begin_Tx
  - manages conflicts; controls Tx aborts using AOU trigger
  - controls commit phase

**FlexWatcher Memory Debugger**
- Extend ISA to support signatures and AOU as first-class entities
- Compiler/Programmer specifies addresses to be tracked
- Hardware triggers on snoops

**Other**
- Buffer Overflow (BO)
  - Pad all heap allocated buffers with 64bytes, watch padded locations
- Memory Leak (ML)
  - Monitor all heap allocated objects and update the address's timestamp on access
- Invariant Violation (IV)
  - Lazy loads cache line for variable X of interest. On AOU handler trigger assert program specific invariants

**RTM [ISCA'07]**
- Integrated Hardware-Software approach to flexible transactional memory
- Hardware Software transactions
  - DIMM mechanisms accelerate common STM operations
  - software makes policy decisions
  - Software routines save uncommon events
  - Realistic Transactional Memory

**Conclusion**
- Data-Isolation and Memory-Monitoring primitives will help multi-core chips achieve widespread use across traditional and emerging application domains
- Decoupling the hardware components will help refine the architecture incrementally and help software evolve the API
- Use simple hardware to accelerate the common case, minimize hardware state and employ software for the uncommon case

Email: {ashriram, sandhya, scott}@cs.rochester.edu